

MICROPROCESSOR

Publication number: JP2002007156 (A)

Publication date: 2002-01-11

Inventor(s): SAEGUSA YASUHIRO

Applicant(s): NIPPON ELECTRIC CO

Classification:

- International: G06F15/78; G06F11/00; G06F15/76; G06F11/00; (IPC1-7): G06F11/00; G06F15/78

• European

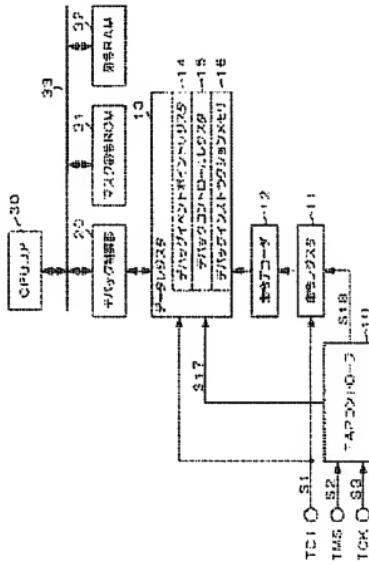
Application number: JP20000186613 20000621

Priority number(s): JP20000186613 20000621

Abstract of JP 2002007156 (A)

PROBLEM TO BE SOLVED: To instantaneously update the erroneous program area of a mask ROM without making it necessary to newly manufacture an LSI, or spending any excessive costs.

SOLUTION: In a data register 13, the start address of the update area of a mask instruction ROM is set through DTI, TMS, and TCK being JTAG terminals in a debug event point register 14, and address break is set in a debug control register 15, and an update program is set in a debug instruction memory 16. When the execution of the instruction of a mask instruction ROM 31 under the control of a CPU core 30 reaches the front of start address of the update area, an update program is executed by a debug control part 20. When the update program is ended, the execution is started from the next program in the update area of the mask instruction ROM 31.



Data supplied from the esp@cenet database — Worldwide

(2) 特開2002-7156 (P2002-7156A)

【特許請求の範囲】

【請求項1】 プログラムが格納されたマスク命令ROMと、該マスク命令ROMのプログラムを実行するCPUコアと、JTAG端子を用いてアクセス可能なデバッグ機能部とを備えるマイクロプロセッサにおいて、

前記デバッグ機能部に、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを設定する設定手段と、

前記CPUコアによるマスク命令ROMのプログラム実行が、前記デバッグ機能部に設定された、更新エリアのスタートアドレスの前まで達すると、前記デバッグ機能部に設定されている更新プログラムを実行させ、更新プログラムが終了すると、前記マスク命令ROMの更新エリアの次のプログラムから実行させるデバッグ制御手段とを具備することを特徴とするマイクロプロセッサ。

【請求項2】 前記スタートアドレスおよび前記更新プログラムは、前記JTAG端子から供給されることを特徴とする請求項1記載のマイクロプロセッサ。

【請求項3】 セルフブートあるいはホストブートにより、外部からインストラクションが設定可能な命令RAMを備え、前記デバッグ機能部が更新プログラムを格納するのに十分な容量がない場合には、更新プログラムの一部を前記命令RAMに格納し、更新プログラムが続続して実行されるようにアドレス設定することを特徴することを特徴とする請求項1記載のマイクロプロセッサ。

【請求項4】 主にデジタル信号処理を行う第1のプロセッサと、主に制御処理を行う第2のプロセッサと、前記第1のプロセッサにより実行されるプログラムが格納されたマスク命令ROMと、JTAG端子を用いてアクセス可能なデバッグ機能部とを備えるマイクロプロセッサにおいて、

前記第2のプロセッサの制御に従って、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを生成する更新制御手段と、

前記更新制御手段により生成された、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを前記デバッグ機能部に設定する設定手段と、

前記CPUコアによるマスク命令ROMのプログラム実行が、前記デバッグ機能部に設定された、更新エリアのスタートアドレスの前まで達すると、前記デバッグ機能部に設定されている更新プログラムを実行させ、更新プログラムが終了すると、前記マスク命令ROMの更新エリアの次のプログラムから実行させるデバッグ制御手段とを具備することを特徴とするマイクロプロセッサ。

【請求項5】 前記JTAG端子からの前記マスク命令ROMの更新エリアのスタートアドレスおよび更新プログラムと、前記更新制御手段からの前記マスク命令ROMの更新エリアのスタートアドレスおよび更新プログラムとのいずれか一方を選択する選択手段を具備し、

前記設定手段は、前記選択手段により選択された、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを前記デバッグ機能部に設定することを特徴とする請求項4記載のマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、JTAG端子を介して内蔵されているデバッグ機能にアクセスし、マスクROMのプログラムを更新するマイクロプロセッサに関する。

【0002】

【従来の技術】 CPUやDSP等のマイクロプロセッサでは、大量生産時にはコスト削減が図られるため、メモリとしてマスクROMを内蔵したマスクROM製品が使用される。また、最近のマイクロプロセッサには、ICEの代わりにデバッグ機能を内蔵し、国際標準規格IEEE1149.1であるバウンダリスキャン(通称JTAG)を使ってデバッグ機能にアクセスするものが増えていている。

【0003】

【発明が解決しようとする課題】 しかしながら、従来技術によるマスクROM製品の場合、LSI製造後にプログラムに誤り(バグ)が発見されると、マスクROMの部品分けを直すために、再度、LSIを製造し直さなければならない。このため、LSIの製造期間および製造費用が余計に発生してしまい、さらには、装置としての正式なリリース時期が遅れ、装置コストにも影響を与える結果となってしまった。

【0004】 この発明は上述した事情に鑑みてなされたもので、新たにLSIを製造する必要も無く、余計な費用を掛けずに、しかも同時に、マスクROMの誤ったプログラムエリアを更新することができるマイクロプロセッサを提供することを目的とする。

【0005】

【課題を解決するための手段】 上述した問題点を解決するために、請求項1記載の発明では、プログラムが格納されたマスク命令ROMと、該マスク命令ROMのプログラムを実行するCPUコアと、JTAG端子を用いてアクセス可能なデバッグ機能部とを備えるマイクロプロセッサにおいて、前記デバッグ機能部に、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを設定する設定手段と、前記CPUコアによるマスク命令ROMのプログラム実行が、前記デバッグ機能部に設定された、更新エリアのスタートアドレスの前まで達すると、前記デバッグ機能部に設定されている更新プログラムを実行させ、更新プログラムが終了すると、前記マスク命令ROMの更新エリアの次のプログラムから実行させるデバッグ制御手段とを備することを特徴とする。

【0006】 また、請求項2記載の発明では、請求項1

記載のマイクロプロセッサにおいて、前記スタートアドレスおよび前記更新プログラムは、前記JTAG端子から供給されることを特徴とする。

【0007】また、請求項3記載の発明では、請求項1記載のマイクロプロセッサにおいて、セルフブートあるいはホストブートにより、外部からインストラクションが設定可能な命令RAMを備え、前記デバッグ機能部が更新プログラムを格納するのに十分な容量がない場合には、更新プログラムの一部を前記命令RAMに格納し、更新プログラムが連続して実行されるようにアドレス設定することを特徴とすることを特徴とする。

【0008】上述した問題点を解決するために、請求項4記載の発明では、主にデジタル信号処理を行う第1のプロセッサと、主に制御処理を行う第2のプロセッサと、前記第1のプロセッサにより実行されるプログラムが格納されたマスク命令ROMと、JTAG端子を用いてアクセス可能なデバッグ機能部とを備えるマイクロプロセッサにおいて、前記第2のプロセッサの制御に従って、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを生成する更新制御手段と、前記更新制御手段により生成された、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを前記デバッグ機能部に設定する設定手段と、前記CPUによるマスク命令ROMのプログラム実行が、前記デバッグ機能部に設定された、更新エリアのスタートアドレスの前まで達すると、前記デバッグ機能部に設定されている更新プログラムを実行させ、更新プログラムが終了すると、前記マスク命令ROMの更新エリアの次のプログラムから実行させるデバッグ制御手段とを具備することを特徴とする。

【0009】また、請求項5記載の発明では、請求項4記載のマイクロプロセッサにおいて、前記JTAG端子からの前記マスク命令ROMの更新エリアのスタートアドレスおよび更新プログラムと、前記更新制御手段からの前記マスク命令ROMの更新エリアのスタートアドレスおよび更新プログラムとのいずれか一方を選択する選択手段を備し、前記設定手段は、前記選択手段により選択された、前記マスク命令ROMの更新エリアのスタートアドレスと更新プログラムとを前記デバッグ機能部に設定することを特徴とする。

【0010】この発明では、JTAG端子を通して、デバッグ機能部へマスク命令ROM31の更新エリア(プログラムとして譲っている部分)のスタートアドレスと、更新プログラムとを設定し、デバッグ制御手段により、前記CPUによるとマスク命令ROMのプログラム実行が、前記デバッグ機能部に設定された、更新エリアのスタートアドレスの前まで達すると、前記デバッグ機能部に設定されている更新プログラムを実行させ、更新プログラムが終了すると、前記マスク命令ROMの更新エリアの次のプログラムから実行させる、したがつ

て、新たにLSIを製造する必要も無く、余計な費用を掛けずに、しかも同時に、マスクROMの誤ったプログラムエリアを更新することが可能となる。

【0011】

【発明の実施の形態】以下、図面を用いて本発明の実施の形態を説明する。

A. 実施形態の構成

図1は、本発明の実施形態による全体の構成を示すブロック図である。図において、TAPコントローラ10は、JTAG端子であるTMSにモード信号S2、TCにクロック信号S3が供給されることにより、命令レジスタ11に制御信号S18を出し、データレジスタ13に制御信号S17を送出する。命令レジスタ11に制御信号S18が供給されると、JTAG端子であるTD1に供給されるデータ信号S1によるインストラクションが設定される。命令デコード12は、命令レジスタ11に設定されたインストラクションをデコードする。データレジスタ13は、デコードされたインストラクションに従って、所望のレジスタを選択する。データレジスタ13の中で選択されたレジスタには、TD1に供給されるデータ信号S1によるデータが制御信号S17により設定される。CPUコア30、デバッグ制御部20、マスク命令ROM31、命令RAM32は、アドレス/データバス33で接続されている。デバッグ制御部20は、CPUコア30がマスク命令ROM31を実行中に、データレジスタ13に設定されたブレーク条件、プログラムを実行する。

【0012】B. 実施形態の動作

次に、上述した実施形態の動作を説明する。ここで、図2は、本実施形態の動作を説明するためのフローチャートである。また、図3は、タイミングチャートである。マイクロプロセッサには、JTAGを利用したデバッグ機能が内蔵されており、それはJTAG端子であるTD1、TMS、TCKと、TAPコントローラ10、命令レジスタ11、命令デコード12、データレジスタ13、デバッグ制御部20とを用いることにより実現される。JTAGは、国際標準規格IEEE1149.1で定められたパワンドリスキャンの通称であり、デバッグ機能はIEEE1149.1でデバイス設計者が独自に決定できるように設けられたプライベート命令を使用する。命令レジスタ11のインストラクション設定、データレジスタ13の選択およびデータ設定は、IEEE1149.1に定められた手順およびインタフェースでTD1、TMS、TCKを通して行われる。

【0013】マスク命令ROM31に、プログラムの読みが発見され、そのメモリエリアがXXアドレスからYYアドレスまでとする。まず、デバッグポイントイベントレジスタ14に更新しなければならないメモリエリアのスタートアドレスXXを設定する(S1)。その後は、図3で示されるように、TD1にデータ信号S1

を、TMSにモード信号S2を、TCKにクロック信号S3を供給することにより、命令レジスタ11にデバッグイベントポイントレジスタを選択するインストラクション(例として20h)を、TAPコントローラ0からの制御信号S18により設定し、それが命令デコード12によりデコードされ、データレジスタ13の中からデバッグイベントポイントレジスタ14が選択される。そして、デバッグポイントレジスタ14には、更新しなければならないメモリエリアのスタートアドレスXXがブレークアドレス値として制御信号S17により設定される。

【0014】同様にして、デバッグコントロールレジスタ15にブレーク要因であるインストラクションアドレスブレークが設定され(S2)、デバッグインストラクションメモリ16に更新プログラムが設定される(S3)。ここで、更新プログラムの容量は、XXXアドレスからYYアドレスまでの領域となるが、本来デバッグ用であるデバッグインストラクションメモリ16には十分なメモリ領域が確保されていない場合がある。デバッグインストラクションメモリ16の領域で足りない場合には、デバッグインストラクションメモリ16の最後尾のインストラクションにYYの次のアドレスへのJUMP命令を設定しておく。デバッグインストラクションメモリ16の領域で足りない場合には、バッファインストラクションの最後尾のインストラクションに命令RAM32のアドレスZZへのJUMP命令を設定しておく。命令RAM32は、マスク命令ROM31に比べて小容量であるが、セルフブートあるいはホストブートにより外部からインストラクションを設定することができる。ここでは、アドレスZZからの領域に予め更新プログラムをブートしておく。命令RAM32にブートする更新プログラムの最後尾のインストラクションにアドレスYYの次のアドレスへのJUMP命令を設定しておく。

【0015】JTAGを通しての設定が終了すると、CPUコア30は、アドレス/データバス30を通してマスク命令ROM31のプログラムを実行していく(S4)。プログラムがアドレスXXの手前まで実行されると、デバッグ制御部20によりインストラクションアドレスブレークが発生し(S5)、デバッグインストラクションメモリ16に設定された更新プログラムが実行される(S6)。更新プログラムがデバッグインストラクションメモリ16の領域に収まっている場合には、デバッグインストラクションメモリ16の最後尾のインストラクション実行後、マスク命令ROM31のアドレスYYの次のアドレスからプログラムが実行される(S7, S8)。一方、更新プログラムがデバッグインストラクションメモリ16の領域に収まっていない場合には、更新プログラムは、デバッグインストラクションメモリ16から命令RAM32に渡り実行され、更新プログラム

が終了すると、マスク命令ROM31のアドレスYYの次のアドレスからプログラムが実行される(S7, S8)。

【0016】C. 他の実施形態

次に、本発明の他の実施形態について説明する。図4は、本発明の他の実施形態における構成を示すブロック図である。なお、図1に対応する部分には同一の符号を付けて説明を省略する。本他の実施形態においては、主にデジタル信号処理を行なうDSPコア34と、マスクロセッサとして主に制御処理を行なうメインCPU50とが搭載された、すなわち1チップ上に複数のプロセッサコアが搭載されている場合を想定している。前述した実施形態においては、更新プログラムは、外部からJTAG端子を通して設定されたが、本他の実施形態では、メインCPU50がプログラム更新コントローラ54を制御することにより行う。

【0017】インストラクションブレークアドレスレジスタ51には、メインCPU50により、アドレス/データバス52を介して、マスク命令ROM31の更新しなければならないメモリエリアのスタートアドレスが設定される。更新プログラムスタートアドレスレジスタ52には、メインCPU50により、アドレス/データバス52を介して、命令RAM32に予めブートされた更新プログラムのスタートアドレスが設定される。セレクタ40は、JTAG端子であるTMS、TCK、TD1を介して外部から入力される信号S2、S3、S1、内部でプログラム更新コントローラ54から生成される信号S56(信号S2に相当)、S57(信号S3に相当)、S58(信号S1に相当)とのいずれか一方、外部端子MUX41から供給される選択信号により選択する。TMS、TCK、TD1はJTAG方式エミュレータと接続し、デバッグを行う場合には、TMS、TCK、TD1を介して外部から供給される信号S2、S3、S1を選択する。一方、DSPコア34およびメインCPU50が実動作中で、マスク命令ROMのあるプログラムエリアを更新したい場合には、プログラム更新コントローラ54で生成される信号S56、S57、S58を選択する。セレクタ40により選択された信号S42(信号S2または信号S56)、信号S43(信号S3または信号S57)は、TAPコントローラ10に、信号S44(信号S1または信号S58)は、命令レジスタ11およびデータレジスタ13に供給される。【0018】セレクタ40がJTAG端子であるTM、TCK、TD1を介して供給される信号S2、S3、S1を出し、マスク命令ROM31の誤ったプログラムエリアを更新する動作については、前述した実施形態と同じであるので説明を省略する。以下では、プログラム更新コントローラ54がJTAG信号に相当する信号S56、S57、S58を用いて、マスク命令ROM31の誤ったプログラムエリアを更新する動作について

て説明する。

【0019】メインCPU50は、アドレス/データバス55を通過してインストラクションブレークアドレスレジスタ51に、マスク命令ROM31の更新しなければならないメモリエリアのスタートアドレスを設定する。同様に、更新プログラムスタートアドレスレジスタ52に、命令RAM32に予めブートされた更新プログラムのスタートアドレスを設定する。メインCPU50がアドレス/データバス55を通過してスタートコマンドレジスタ53にアクセスすると、プログラム更新コントローラ54では、JTAG信号である信号S56、S57、S58が生成される。

【0020】次に、セレクタ40により、プログラム更新コントローラ54からの信号S56、S57、S58が選択され、信号S56、S57が信号S42、S43として、TAPコントローラ10に供給され、信号S58が信号S44として、命令レジスタ11およびデータレジスタ13に供給される。データレジスタ13では、デバッグイベントポイントレジスタ14に更新しなければならないメモリエリアのスタートアドレスが設定され、デバッグコントロールレジスタ15にインストラクションアドレスブレークが設定され、デバッグインストラクションメモリ16に命令RAM32に予めブートされた更新プログラムのスタートアドレスへのJUMP命令が設定される。以下、前述した実施形態と同様の動作となる。

【0021】上述した他の実施形態は、前述した実施形態がチップの外部であるボード上にプログラムを更新するための制御ハードウェアを設ける必要があったのに対し、複数のプロセッサを搭載しているチップにおいては、一方がマスターとして制御することにより、外部ボード上に制御ハードウェアを設ける必要がなく、任意にマスクROMの内容を更新できる。

【0022】

【発明の効果】以上説明したように、本発明によれば、JTAGを利用して本来デバッグ機能として内蔵されている機能ブロックをアクセスすることにより、大量生産時にはコスト削減がはかれるマスクROMを内蔵したマイクロプロセッサにおいても、プログラムの読み(バ

グ)に対し、新たにLSIを製造する必要も無く、余計な費用を掛けずに、しかも瞬時にプログラムを更新できるという利点が得られる。また、JTAG機能としてのJTAG端子およびハードウェア資源は本来マイクロプロセッサのオンチップデバッグ機能として内蔵されているため、共有化が図れ、余分な端子、ハードウェアを設ける必要がなく、共有化できない場合と比べ開発工数を少なくすることができ、さらにはチップ単価を安くすることができるという利点が得られる。

【図面の簡単な説明】

【図1】 本発明の実施形態による全体の構成を示すブロック図である。

【図2】 本実施形態の動作を説明するためのフローチャートである。

【図3】 本実施形態の動作を説明するためのタイミングチャートである。

【図4】 本発明の他の実施形態における構成を示すブロック図である。

【符号の説明】

1.0 TAPコントローラ (設定手段)

1.1 命令レジスタ (設定手段)

1.2 命令デコード (設定手段)

1.3 データレジスタ (デバッグ機能部)

1.4 デバッグイベントポイントレジスタ

1.5 デバッグコントロールレジスタ

1.6 デバッグインストラクションメモリ

2.0 デバッグ制御部 (デバッグ制御手段)

3.0 CPUコア

3.1 マスク命令ROM

3.2 命令RAM

3.3 アドレス/データバス

3.4 DSPコア (第1のプロセッサ)

4.0 セレクタ (選択手段)

5.0 メインCPU (第2のプロセッサ)

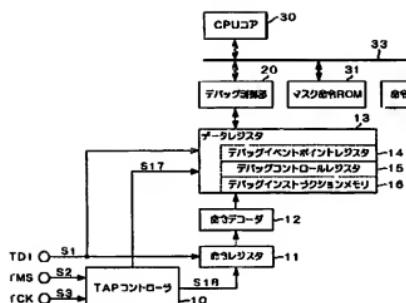
5.1 インストラクションブレークアドレスレジスタ

5.2 更新プログラムスタートアドレスレジスタ

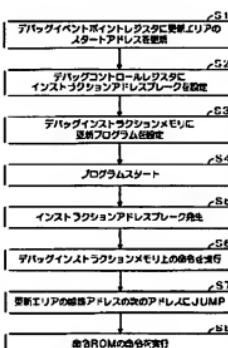
5.3 スタートコマンドレジスタ

5.4 プログラム更新コントローラ

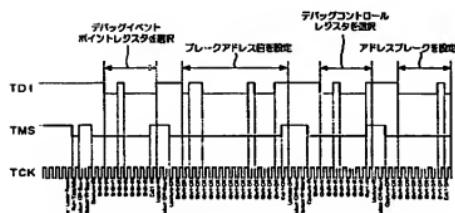
11



【图2】



【图3】



【四】

